WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Tuesday, June 01, 2004

Hide?	Set Name	Query	Hit Count
		DB=USPT,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR	
☐	L7	l3 same l5	0
☐	L6	l4 and L5	4
☐	L5	(first near2 (system or computer)) same (second (system or computer))	205589
☐	L4	L3 same creat$7	25
☐	L3	(deploy$7 near3 configur$7) same install$7	141
☐	L2	L1 same install$7	0
☐	L1	(deploy$7 near3 configur$7 near3 (program$4 or software))	40

END OF SEARCH HISTORY

L6: Entry 2 of 4 File: USPT Sep 3, 2002

DOCUMENT-IDENTIFIER: US 6446260 B1
TITLE: Method and apparatus for operating system personalization during installation

Abstract Text (1):
A method an apparatus for providing personalization parameters to allow an operating system to install itself on a computer system with the provided personalization parameters. A first operating system personalization file is used by the operating system to configure itself. A personalization parameters file is provided and read by an operating system installation process which is initiated in the computer system. The operating system installation process then displays a graphical user interface from which a user to selects personalization parameters with which the operating system is to configure itself. An editing module is executed to edit the operating system configuration file to include at least a portion of the selected personalization parameters so that the operating system is configured with the selected personalization parameters when it installs itself.

Brief Summary Text (16):
The present invention overcomes the above described disadvantages by providing an economical, high performance and adaptable apparatus and method for providing personalization parameters for an automated operating system installation on a computer system. The computer system can include a memory, a first CPU, and a second CPU interconnected with the first CPU to allow the operating system to be downloaded from the first CPU to the second CPU. The method includes providing both an operating system configuration file and a personalization parameters file in the memory. The operating system configuration file is used by the operating system to configure itself with personalization parameters contained in the file. On a display element of the second CPU, a graphical user interface ("GUI") is displayed which lists the personalization parameters available in the personalization parameters file. The GUI also allows a user to select a plurality of the personalization parameters from the personalization parameters file. The operating system configuration file is edited by an editing module to include at least a first portion of the selected plurality of personalization parameters. In this way, the operating system can install itself configured with the first portion of the selected plurality of personalization parameters.

Brief Summary Text (17):
In another aspect of the method for providing personalization parameters for an automated operating system installation from a first CPU to a second CPU, the computer system is provided with operating system application programming interfaces ("APIs") each of which is for configuring an operating system with a personalization parameter. After the operating system has installed itself, the operating system APIs are edited to each include one of a second portion of the selected plurality of personalization parameters. The plurality of APIs are executed to configure the operating system with the second portion of the selected plurality of personalization parameters.

Brief Summary Text (18):
Another aspect of the present invention includes an apparatus for automatically providing personalization parameters for a personalized operating system

installation in a computer system. The computer system includes a memory, a first CPU, and a second CPU interconnected with the first CPU to allow the operating system to be downloaded from the first CPU to the second CPU, the second CPU also having a display element. The apparatus includes an operating system configuration file for storing personalization parameters with which the operating system configures itself during an installation. The apparatus also includes a personalization parameters file stored in the memory. A GUI is displayed on the display element of the second CPU. The GUI allows a user to select a plurality of the personalization parameters in the personalization parameters file. A first editing module places a first portion of the personalization parameters selected via the GUI from the personalization parameters file into the operating system configuration file. In this way, when the operating system is installed, it configures itself with the first portion of the personalization parameters.

Brief Summary Text (19):
In another aspect of an apparatus for automatically providing personalization parameters for an operating system installation from a first CPU to a second CPU, the apparatus includes a plurality of operating system APIs stored in the memory. Each API is for configuring the operating system with one personalization parameter after installation of the operating system. The apparatus also includes a second editing module and a post-operating system installation module. The second editing module is for placing one of a second portion of the selected personalization parameters from the personalization parameters file into each of the APIs. The post-operating system installation module executes the APIs to configure the operating system with the second portion personalization parameters.

Brief Summary Text (20):
If, for example, the first operating system configuration file is a Microsoft answer file, then the present invention advantageously avoids the need for a system administrator to create an answer file containing end-user specific personalization parameters for each workstation on a network. Rather, the system administrator must only create a single "generic" answer file (that is, an answer file containing parameters that can be used by all the computers on network or a group of computers on a network) and a personalization parameters file. The method and apparatus of the present invention then allows selection of personalization parameters from the personalization parameters file and automatically edits the generic answer file to create a user-specific, updated answer file. The operating system then uses this updated answer file to configure itself during installation.

Detailed Description Text (32):
After display of the GUI and selection of a group of personalization parameters with which the deployed operating system is to be configured, an editing module of the present invention places the personalization parameters from the selected group in generic answer file 64 to create an updated answer file 66. FIGS. 7 and 8 illustrate the steps of this editing process. Initially, in step 134, an editing module, referred to as an "OS installation module" 54, reads the value of comprofiles environment variable 95 and then, in step 136 reads the value of selectedcomp environment variable 97. In steps 138 and 140, OS installation module 54 opens the desktop profile file 34 designated in comprofiles and retrieves the attributes of the selected desktop profile 39 designated in selectedcomp. In step 142, OS installation module retrieves and updates generic answer file 64 with the parameters identified in the selected desktop profile.

L6: Entry 4 of 4 File: USPT Dec 7, 1999

DOCUMENT-IDENTIFIER: US 5999972 A
TITLE: System, method and article of manufacture for a distributed computer system framework

Detailed Description Text (155):
An additional development task is to modify the server program to specify which function to call in response to inbound messages and make calls to a server Communication Library to send results to the client. All of the components in an ICE-T application system reside on the server. To deploy an application, you install its components and additional ICE-T files and programs that manage applications on the server. ICE-T also provides a template for creating a startup applet that enables users to start applications from a browser. Chapter 3, "Configuring and Deploying ICE-T Applications" described these tasks and tools.

CLAIMS:

1. A distributed computer system, comprising:

(a) a client computer code segment resident on a client computer node, the client computer code segment containing a first portion of executable code of a single application program and a first set of handlers for handling a message set specific to the single application program, the client computer code segment initiated on the client computer node;

(b) a server computer code segment resident on a server computer node coupled to the client computer node, the server computer code segment containing a second portion of executable code of the single application program and a second set of handlers for handling the message set specific to the single application program, the server computer-code segment initiated on the client computer node, such that the first portion of executable code and the second portion of executable code combined make up the single application program and such that the single application program can only be executed by commands from the client computer node;

(c) an execution framework code segment configured to couple the client computer code segment and the server computer code segment using the message set specific to the single application program to facilitate event driven message transfer between the client computer code segment and the server computer code segment, thereby enabling execution of the single application program in a distributed manner between the client computer node and the server computer code.

8. A method for distributing computer between a server computer system and a client computer system, comprising the steps of:

(a) storing a client computer code segment resident on a client computer node, the client computer code segment containing a first portion of executable code of a single application program and a first set of handlers for handling a message set specific to the single application program, the client computer code segment

initiated on the client computer node;

(b) storing a server computer segment resident on a server computer node coupled to the client computer node, the server computer code segment containing a second portion of executable code of the single application program and a second set of handlers for handling the message set specific to the single application program, the server computer code segment initiated on the client computer node, such that the first portion and the second portion combined make up the single application program and such that the single application program can only be executed by commands from the client computer node;

(c) configuring an execution framework code segment to couple the client computer code segment and the server computer code segment using the message set specific to the single application program to facilitate event driven message transfer between the client computer code segment and the server computer code segment, thereby enabling execution of the single application program in a distributed manner between the client computer node and the server computer node.

15. A computer program embodied on a computer-readable medium for enabling a distributed computer system, comprising:

(a) a client computer code segment resident on a client computer node, the client computer code segment containing a first portion of executable code of a single application program and a first set of handlers for handling a message set specific to the single application program, the client computer code segment initiated on the client computer node;

(b) a server computer code segment resident on a server computer node coupled to the client computer node, the server computer code segment containing a second portion of executable code of the single application program and a second set of handlers for handling the message set specific to the single application program, the server computer code segment initiated on the client computer node, such that the first portion and the second portion combined make up the single application program and such that the single application program can only be executed by commands from the client computer code;

(c) an execution framework code segment configured to couple the client computer code segment and the server computer code segment using the message set specific to the single application program to facilitate event driven message transfer between the client computer code segment and the server computer code segment, thereby enabling execution of the single application program in a distributed manner between the client computer node and the server computer node.